

# Generating ElGamal Signatures Without Knowing the Secret Key

Daniel Bleichenbacher

ETH Zürich

Institute for Theoretical Computer Science

CH-8092 Zürich, Switzerland

email: [bleichen@inf.ethz.ch](mailto:bleichen@inf.ethz.ch)

**Abstract.** We present a new method to forge ElGamal signatures if the public parameters of the system are not chosen properly. Since the secret key is hereby not found this attack shows that forging ElGamal signatures is sometimes easier than the underlying discrete logarithm problem.

## 1 Introduction

ElGamal's digital signature scheme [4] relies on the difficulty of computing discrete logarithm in the multiplicative group  $\mathbb{F}_p^*$  and can therefore be broken if the computation of discrete logarithms is feasible. However, the converse has never been proved. In this paper we show that it is sometimes possible to forge signatures without breaking the underlying discrete logarithm problem. This shows that the ElGamal signature scheme and some variants of the scheme must be used very carefully.

The paper is organized as follows. Section 2 describes the ElGamal signature scheme. In Section 3 we present a method to forge signatures if some additional information on the generator is known. We show that signatures can be forged if the generator  $\alpha$  is smooth and divides  $p - 1$ . Hence for example  $\alpha = 2$  is a totally insecure choice. In Section 4 we discuss the case where the public parameters are not chosen by the user itself. The authority that chooses these parameters may generate some trapdoor information, which will allow it to sign arbitrary messages for any user. Section 5 contains a brief description of some possible countermeasures. In Section 6 we discuss a variant of the ElGamal scheme over  $\mathbb{Z}/n\mathbb{Z}$  with  $n = pq$ , which was believed to be as hard as factoring and computing discrete logarithms. However, the factorization of  $n$  can often be derived from known signatures. Moreover, we show that in this case computation in only one of the groups  $\mathbb{F}_p^*$  or  $\mathbb{F}_q^*$  is sufficient to forge signatures. Again it is not necessary to discover the complete secret key of a user to generate signatures.

## 2 ElGamal's signature scheme

ElGamal's signature scheme can be described as follows.

*Public parameters.* Let  $p$  be a large prime and  $\alpha$  a generator of the multiplicative group  $\mathbb{F}_p^*$ .

*Secret key and public key.* Every user  $A$  chooses a secret key  $x_A \in \{0, \dots, p-2\}$  and publishes  $y_A \equiv \alpha^{x_A} \pmod{p}$  as his public key.

*Computation of a signature.* Let  $0 \leq h < p-1$ . To compute a signature on  $h$  user  $A$  proceeds as follows. First he chooses a random value  $k \in \{0, \dots, p-2\}$  relatively prime to  $p-1$  and computes  $1 \leq r < p$  and  $s$  by

$$\begin{aligned} r &\equiv \alpha^k \pmod{p} \\ \text{and } s &\equiv (h - x_A r)k^{-1} \pmod{p-1}. \end{aligned}$$

The pair  $(r, s)$  is a valid signature on  $h$ .

*Verification of a signature.* Any user knowing the public key  $y_A$  can verify the signature by checking that  $1 \leq r < p$  and the following equation are satisfied.

$$\alpha^h \equiv r^s y_A^r \pmod{p}$$

The ElGamal signature scheme can be broken when discrete logarithms in  $\mathbb{F}_p^*$  can be computed. The prime  $p$  must therefore be chosen large enough to prevent the computation of discrete logarithms by the number field sieve [6] and  $p-1$  must contain at least one large prime factor to disable the algorithm of Pohlig and Hellman [13]. The value  $h$  that occurs in the signature is normally not equal to the message to sign, it is rather the result of a collision free hash function applied to the message. This avoids the existential attack described in [4]. It is important that the verifier checks whether  $1 \leq r < p$  is satisfied. If he would accept signatures where  $r$  is larger than  $p$  then any signature  $(r, s)$  on  $h$  could be used to generate a signature  $(r_2, s_2)$  on an arbitrary hash value  $h_2$  by setting  $u \equiv h_2 h^{-1} \pmod{p-1}$ . This implies

$$\alpha^{h_2} \equiv \alpha^{hu} \equiv (y_A)^{ru} r^{su} \pmod{p}.$$

Now  $(r_2, s_2)$  can be found by setting  $s_2 \equiv su \pmod{p-1}$  and by computing  $r_2$  satisfying  $r_2 \equiv ru \pmod{p-1}$  and  $r_2 \equiv r \pmod{p}$  by using the Chinese Remainder Theorem. This kind of attack will for example be used in Section 6.

### 3 Weak generators

The security of the ElGamal signature scheme depends heavily on the parameters  $p$  and  $\alpha$ . The following theorem shows that ElGamal signatures can be generated when some additional information on the generator  $\alpha$  is available.

**Theorem 1.** *Let  $p - 1 = bw$  where  $b$  is smooth and let  $y_A$  be the public key of user  $A$ . If a generator  $\beta = cw$  with  $0 < c < b$  and an integer  $t$  are known such that  $\beta^t \equiv \alpha \pmod{p}$  then a valid ElGamal signature  $(r, s)$  on a given  $h$  can be found.*

*Proof.* First we show that the equation

$$\alpha^{wz} \equiv (y_A)^w \pmod{p}$$

can be solved for  $z$ . It follows from  $p - 1 = bw$  that the subgroup  $H$  generated by  $\alpha^w$  has order  $b$ . Since  $b$  is smooth it is possible to compute discrete logarithms in  $H$  by using the algorithm of Pohlig and Hellman [13], so that  $z$  can be found. Now let

$$\begin{aligned} r &= \beta \\ s &\equiv t(h - cwz) \pmod{p - 1}. \end{aligned}$$

Then  $(r, s)$  is a valid signature on  $h$  since

$$\begin{aligned} r^s (y_A)^r &\equiv (\beta^t)^{(h - cwz)} (y_A)^{cw} \\ &\equiv \alpha^{h - cwz} \alpha^{cwz} \equiv \alpha^h \pmod{p}. \quad \square \end{aligned}$$

**Corollary 2.** *If  $\alpha$  is smooth and divides  $p - 1$  then it is possible to generate a valid ElGamal signature on an arbitrary value  $h$ .*

*Proof.* Let  $\beta = (p - 1)/\alpha$  and  $t = (p - 3)/2$ . Then  $\beta^t \equiv (-1)\beta^{-1} \equiv \alpha \pmod{p}$ . Thus it follows by Theorem 1 that signatures for all  $h$  can be forged.  $\square$

Thus if the generator  $\alpha$  is chosen badly then signatures on every given message can be found without knowing the secret key. Choosing  $\alpha = 2$  is exceptionally bad since it allows to forge signatures independently of the choice of  $p$ . Such small generators are sometimes chosen in order to get an efficient exponentiation.

It should be noted that this attack succeeds because of the special choice of  $r$ , which reduces the discrete logarithm problem in  $\mathbb{F}_p^*$  to the discrete logarithm problem in a subgroup of  $\mathbb{F}_p^*$  with smooth order. Another attack that is based on the fact that discrete logarithms in small subgroups are computable has been described recently by Menezes Qu and Vanstone [11]. (Their attack shows that authenticity is not guaranteed in a few Diffie-Hellman based key agreement protocols.)

## 4 Constructing a trapdoor

The public parameters  $p$  and  $\alpha$  may be fixed so that every user in the system uses the same group and generator. Such a convention is attractive because it allows the use of shorter public keys. Additionally, signatures can be computed and verified much faster by using precomputed exponents [2]. However, the authority choosing the prime  $p$  and the generator  $\alpha$  for a signature system may additionally generate some trapdoor information that can be used to forge arbitrary messages later.

One way to generate a prime  $p$  such that  $\mathbb{F}_p^*$  has a trapdoor has been pointed out in [1, p.50]. The prime  $p$  can be generated together with a polynomial that is highly suitable for the number field sieve. The authority will then be able to compute discrete logarithms faster than a user who does not know the trapdoor [5]. However, this gives only a moderate advantage and can be avoided by choosing the prime  $p$  sufficiently large. Moreover, such primes can be recognized fairly easily [14].

Here we present another way to generate a trapdoor. An authority that can choose the public parameters  $p$  and  $\alpha$  can generate these parameters such that it additionally knows secret values  $\beta$  and  $t$  satisfying the constraints of Theorem 1. We illustrate this possibility by giving two different methods to generate the trapdoor. The first method shows how a generator  $\alpha$  and the trapdoor information can be generated given a fixed prime  $p$ . The second method shows how a prime  $p$  and the trapdoor information can be generated given a fixed small generator  $\alpha$ . There is no guarantee that these methods succeed. However, the probability of success is sufficiently high to threaten the security of the system.

*Method A.* Let  $p - 1 = bw$  with  $b$  smooth. If  $b$  is not too small then we can find  $\alpha, \beta$  and  $t$  in the following way. We choose  $c \in \{1, \dots, b - 1\}$  randomly until  $\beta = cw$  is a generator of  $\mathbb{F}_p^*$ . Finally we chose  $t$  with  $\gcd(t, p - 1) = 1$  and compute  $\alpha = \beta^t$ .

This attack is practical when a generator  $\beta = cw$  of  $\mathbb{F}_p^*$  can be found in reasonable time. When  $b$  is too small then no generator of the form  $cw$  may exist. However, the number of generators of  $\mathbb{F}_p^*$  is  $\varphi(p - 1)$ . Since  $n/\varphi(n) = O(\ln \ln(n))$  (see for example [7]) we expect to find a generator after  $O(\ln \ln(p))$  trials.

When  $t$  is chosen uniformly from the set of all  $1 \leq t < p - 1$  that are relatively prime to  $p - 1$  then  $\alpha = \beta^t$  is a random generator of  $\mathbb{F}_p^*$ . Thus it is impossible to detect the trapdoor as long as no false signature has been published. However, the trapdoor can be reconstructed from a given false signature  $(\beta, s)$  on  $h$  since  $\alpha^h \equiv \beta^s \pmod{p}$  implies  $h \equiv ts \pmod{w}$ . In order to find  $t$  it is then sufficient to compute  $\log_\alpha(\beta) \pmod{b}$  and this can be done efficiently as  $b$  is smooth.

*Method B.* When the generator  $\alpha$  is fixed then  $p, \beta$  and  $t$  can be generated as follows. First three positive integers  $u, v, c$  are selected such that  $v$  is odd and  $c^v \alpha^u$  has approximately the size of the prime to construct. Next the smooth divisors of  $c^v \alpha^u - 1$  are computed. This can be done easily using for example trial division or Pollard-rho factorization since only the small prime factors of

$c^v \alpha^u - 1$  are wanted. If there exists a smooth divisor  $d > c$  of  $c^v \alpha^u - 1$  such that  $p = c^v \alpha^u - d^v$  is prime,  $\frac{p-1}{2} - u$  is relatively prime to  $p-1$  and  $\alpha$  is a generator of  $\mathbb{F}_p^*$  then it is possible to construct a trapdoor as follows.

$$\beta = c \frac{p-1}{d}$$

$$t \equiv v \left( \frac{p-1}{2} - u \right)^{-1} \pmod{p-1}$$

It follows from  $d | (c^v \alpha^u - 1)$  that  $d$  divides  $p-1$ . Thus  $\beta$  satisfies the precondition of Theorem 1. Because of  $\alpha^u c^v \equiv d^v \pmod{p}$  we have  $\alpha^{-u} \equiv (cd^{-1})^v \pmod{p}$  and therefore  $\beta^v \equiv (-cd^{-1})^v \equiv (-1)\alpha^{-u} \equiv \alpha^{(p-1)/2-u} \pmod{p}$ . Hence  $\beta^t \equiv \alpha \pmod{p}$ . It follows that Theorem 1 can be applied.

A heuristic analysis shows that the runtime of this method depends on the size of  $c$  and should find a trapdoor after trying  $O((\ln p)(\ln \ln p)^2)$  values for  $p$ . An example for such a trapdoor, which has been generated in less than a day on a SPARC IPC, is given by the values  $\alpha = 5, u = 227, v = 1, c = 1629$  and  $d = 2^2 \cdot 7 \cdot 23 \cdot 47 \cdot 78541 \cdot 3489781$ .

## 5 Countermeasures

The attacks shown in Section 3 and 4 can be avoided if signatures  $(r, s)$  are considered to be valid only if — additionally to the other conditions —  $r$  is not divisible by a large prime divisor  $q$  of  $p-1$ . This condition should always be checked by the verifier. Moreover, an authorized signer will almost always generate a valid signature since it is very unlikely that he randomly generates an  $r$  that is divisible by  $q$ . Such a condition has been included in the digital signature standard (DSS) [12]. Hence the DSS is not susceptible to the attacks presented in this paper.

Alternatively trapdoors may be avoided if the authority that is choosing the public parameters  $p$  and  $\alpha$  is forced to use an algorithm like the one proposed by NIST for the generation of  $p$  in DSS. The values produced by this algorithm allow to verify publicly that the parameters have indeed been generated by the algorithm. This would make it very hard for a dishonest authority to create a trapdoor. The two methods to generate trapdoors shown in Section 4 indicate that both  $p$  and  $\alpha$  must be generated with this algorithm if no other steps to prevent the attacks are taken.

Yet another possibility to avoid the attacks might be to modify the equations for signature generation and verification (see for example [9] for an overview of ElGamal variants). Such a variant must be chosen carefully since other problems may arise. For example if a signature on  $h$  is computed by  $r \equiv \alpha^k \pmod{p}$  and  $s \equiv x_A + hkr \pmod{p-1}$  and therefore verified by  $\alpha^s \equiv y_A r^{hr} \pmod{p}$  then a chosen message attack is possible if the signer can be forced to sign a message  $h$  where  $\gcd(p-1, h)$  is large. Any such signature leaks information about the secret key  $x_A$  since  $s \equiv x_A + hkr \pmod{p-1}$  implies  $s \equiv x_A \pmod{\gcd(p-1, h)}$ . A special case of this attack has been discussed in [9].

## 6 An ElGamal scheme over $\mathbb{Z}/n\mathbb{Z}$

In [15] Saryazdi has proposed a variant of the ElGamal signature system using  $(\mathbb{Z}/n\mathbb{Z})^*$  where  $n$  is the product of two large primes  $p$  and  $q$ . The author hopes that the security of the proposed signature system relies on the factoring problem and the discrete logarithm problem. Moreover, the existential attack shown in [4] seems not possible in that scheme, because this attack requires that the order of the group  $(\mathbb{Z}/n\mathbb{Z})^*$  is known. However, Horster et al. observed in [10] that the signatures leak information that can be used to factor the modulus. Even though their attack does not work as described it is possible to modify the attack in such a way that it works for Saryazdi's scheme as well as the improved scheme proposed in [10]. Moreover, we show that computing discrete logarithms in only one of the two groups  $\mathbb{F}_p^*$  or  $\mathbb{F}_q^*$  is sufficient to break the scheme.

*Description of the scheme.* In Saryazdi's variation of the ElGamal signature scheme every user  $A$  chooses two large primes  $p$  and  $q$  and computes  $n = pq$ . Then she tries to find an integer  $\alpha \in (\mathbb{Z}/n\mathbb{Z})^*$  with order  $\lambda(n)$ . Furthermore he chooses a random element  $x_A$  and computes  $y_A = \alpha^{x_A} \pmod{n}$ .  $p, q$  and  $x_A$  are kept secret whereas  $n, \alpha$  and  $y$  are published as  $A$ 's public key. To sign  $h \in \mathbb{Z}/n\mathbb{Z}$  user  $A$  chooses a random number  $k \in (\mathbb{Z}/n\mathbb{Z})^*$  and computes

$$r \equiv \alpha^k \pmod{n} \quad (1)$$

$$s \equiv (h - x_A r) k^{-1} \pmod{\varphi(n)} \quad (2)$$

Then  $(r, s)$  is the signature on  $h$  where  $h$  is either the message or the hash value of a message. A verifier accepts a signature  $(r, s)$  on  $h$  if  $1 \leq r < n$  and

$$\alpha^h \equiv (y_A)^r r^s \pmod{n}.$$

This scheme does not allow a modulus  $n$  that is common to all users, since every user has to know  $\varphi(n)$  in order to be able to compute signatures.

*Description of the attack.* Assume that  $t$  is a small prime that divides  $p - 1$  but not  $q - 1$ . Assume further that  $(r_i, s_i) : 1 \leq i \leq d$  are known signatures on  $h_i$  such that  $t$  divides  $s_i$  and such that the system

$$\sum_{i=1}^d c_i h_i \equiv 0 \pmod{t} \quad (3)$$

$$\text{and } \sum_{i=1}^d c_i r_i \equiv 0 \pmod{t} \quad (4)$$

has a nontrivial solution (i.e.  $c_i \not\equiv 0 \pmod{t}$  for at least one  $i$ ) in the coefficients  $c_i$ . Such a solution to (3) and (4) can always be found when at least 3 signatures with  $t|s_i$  are known. If we set  $B := \sum_{i=1}^d c_i h_i$  and  $C := \sum_{i=1}^d c_i r_i$  then we have

$$\alpha^B \equiv (y_A)^C \prod_{i=1}^d r_i^{c_i s_i} \pmod{n}. \quad (5)$$

Since  $t$  divides all exponents in (5) we can compute the gcd of  $n$  and

$$\alpha^{B/t} - (y_A)^{C/t} \prod_{i=1}^d r_i^{c_i s_i/t}.$$

This will factor  $n$  if exactly one of the following two equations are satisfied.

$$\alpha^{B/t} \equiv (y_A)^{C/t} \prod_{i=1}^d r_i^{c_i s_i/t} \pmod{p} \quad (6)$$

$$\alpha^{B/t} \equiv (y_A)^{C/t} \prod_{i=1}^d r_i^{c_i s_i/t} \pmod{q} \quad (7)$$

Since  $t$  is relatively prime to  $q - 1$  it follows from (5) that (7) is satisfied. Since  $\alpha$  is a generator modulo  $p$  it follows that (6) is satisfied if and only if

$$B/t \equiv x_A(C/t) + \sum_{i=1}^d k_i c_i (s_i/t) \pmod{p-1}$$

or equivalently

$$B \equiv x_A C + \sum_{i=1}^d k_i c_i s_i \pmod{t(p-1)} \quad (8)$$

is satisfied. From (2) follows only  $B \equiv x_A C + \sum_{i=1}^d k_i c_i s_i \pmod{\varphi(n)}$ . Let  $t^z$  be the maximal power of  $t$  dividing  $\varphi(n)$ . It follows that  $t^z$  divides  $p-1$  and hence  $t^{z+1}$  divides  $t(p-1)$ . Thus we expect that (8) is only satisfied with probability about  $1/t$ .

It should be noted that an attacker does not know which primes  $t$  may be successful since he does not know  $p$  and  $q$  and can generally not determine whether a given  $t$  divides  $p-1$  but not  $q-1$ . But this is not a big problem because an attacker can simply try all small primes  $t$  that are a divisor of some  $s_i$ 's.

If the factorization of  $n$  can be discovered then signatures can be found if computing discrete logarithms in  $\mathbb{F}_p^*$  but not necessarily in  $\mathbb{F}_q^*$  is feasible, provided that  $p$  is not much smaller than  $q$ . This can be done as follows.

We may assume that the attacker knows a valid signature  $(r, s)$  on  $h$ . This may be a previously given signature or one that was generated using the existential forgery described in [4]. If he wants to generate a signature on  $h'$  then he uses the Chinese Remainder Theorem to compute  $0 \leq r' < q(q-1)$  such that

$$\begin{aligned} r' &\equiv r h^{-1} h' \pmod{q-1} \\ r' &\equiv r \pmod{q} \end{aligned}$$

If  $p$  is not much smaller than  $q$  then he will find an  $r' < n$  in reasonable time. If we set  $s_q \equiv s h^{-1} h' \pmod{q-1}$  then we have

$$\alpha^{h'} \equiv (y_A)^{r'} r'^{s_q} \pmod{q}$$

Assuming that the discrete logarithm in  $\mathbb{F}_p^*$  is feasible it is possible to solve

$$h' \equiv \log_\alpha(y_A)r' + \log_\alpha(r')s_p \pmod{p-1}$$

for  $s_p$  if  $\log_\alpha(r')$  is relatively prime to  $p-1$ . If additionally the system

$$s' \equiv s_p \pmod{p-1}$$

$$s' \equiv s_q \pmod{q-1}$$

is solvable then  $(r', s')$  is a valid signature on  $h'$ .

*Remarks.* If  $t$  divides both  $p-1$  and  $q-1$  then it follows from (2) that

$$B \equiv x_A C + \sum_{i=1}^d k_i c_i s_i \pmod{\varphi(n)}$$

is satisfied. This implies that (8) and the analogous equation modulo  $q-1$  are satisfied. The attack does therefore not work when  $t$  divides  $p-1$  and  $q-1$ . The attack described in [10] considers the case  $t=2$  only and does not work therefore. However, it would work when equation (2) is reduced modulo  $\lambda(n)$  instead of  $\varphi(n)$  provided that the maximal powers of 2 in  $p-1$  and  $q-1$  are not the same. The authors of [10] propose that only signatures  $(r, s)$  where  $s$  is odd are allowed. This proposal does not prevent the attack described here since  $s$  can still be divisible by a small prime divisor  $t$ .

Another consequence of the note above is that the attack does not work when the smooth parts of  $p-1$  and  $q-1$  are equal. Thus the attack presented here can be avoided by choosing  $p-1$  and  $q-1$  such that  $(p-1)/2$  and  $(q-1)/2$  are prime, since then no small prime  $t$  exists dividing  $p-1$  but not  $q-1$ . Some of the extensions proposed in [10] seem to avoid the attack presented here too. Other variants of ElGamal based on the discrete logarithm problem and the factoring problem that are more practical than Saryazdi's scheme have been proposed by Brickell and McCurley [3] and Harn [8].

## 7 Conclusion

The results presented in this paper show that ElGamal signatures can be forged in some cases without knowing the secret key. The attacks presented can be avoided by restricting the values of signatures that are considered to be valid. The ElGamal digital signature scheme has therefore not been broken but it has been shown that the system must be used very carefully.

## Acknowledgments

I'm grateful to Ueli Maurer, Markus Stadler, Holger Petersen Markus Michels and some members of the EUROCRYPT program committee for their comments and suggestions.



## References

1. T. Beth, M. Frisch, and G.J. Simmons (eds). *Public-key Cryptography, State of the Art and Future Directions*, volume 578 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 1992.
2. E. F. Brickell, D. M. Gordon, K. S. McCurley, and D. B. Wilson. Fast exponentiation with precomputation. *Advances in Cryptology - EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 200–207, 1993.
3. E. F. Brickell and K. S. McCurley. An interactive identification scheme based on discrete logarithms and factoring. *Journal of Cryptology*, 5(1):29–39, 1992.
4. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Advances in Cryptology: Proceedings of CRYPTO '84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer-Verlag, 1985.
5. D. M. Gordon. Designing and detecting trapdoors for discrete log cryptosystems. *Advances in Cryptology - CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 66–75. Springer-Verlag, 1992.
6. D. M. Gordon. Discrete logarithms in  $GF(p)$  using the number field sieve. *SIAM J. Disc. Math.*, 6(1):124–138, February 1993.
7. G. H. Hardy and E. M. Wright. *An introduction to the theory of numbers*. Clarendon Press, Oxford, 5th edition, 1979.
8. L. Harn. Public-key cryptosystem design based on factoring and discrete logarithm. *IEE Proc. Comput. Digit. Tech.*, 141(3):193–195, 1994.
9. P. Horster, M. Michels, and H. Petersen. Generalized ElGamal signatures for one message block. Technical Report TR-94-3, University of Technology Chemnitz-Zwickau, May 1994.
10. P. Horster, M. Michels, and H. Petersen. Meta-ElGamal signature schemes using a composite module. Technical Report TR-94-16-E, University of Technology Chemnitz-Zwickau, November 1994.
11. A. Menezes, M. Qu, and S. Vanstone. Key agreement and the need for authentication. PKS, November 1995.
12. National Institute of Standards and Technology (NIST). *FIPS Publication 186: Digital Signature Standard*, May 19, 1994.
13. S. C. Pohlig and M. E. Hellman. An improved algorithm for computing logarithms over  $GF(p)$  and its cryptographic significance. *IEEE Trans. Inform. Theory*, IT-24:106–110, January 1978.
14. R. A. Rueppel, A. K. Lenstra, M. E. Smid, K. S. McCurley, Y. Desmedt, A. Odlyzko, and P. Landrock. Panel discussion: Trapdoor primes and moduli. *Advances in Cryptology — EUROCRYPT '92*, volume 658 of *Lecture Notes in Computer Science*, pages 194–199. Springer-Verlag, 1993.
15. S. Saryazdi. An extension to ElGamal public key cryptosystem with a new signature scheme. *Communication, Control, and Signal Processing*, pages 195–198. Elsevier, 1990.